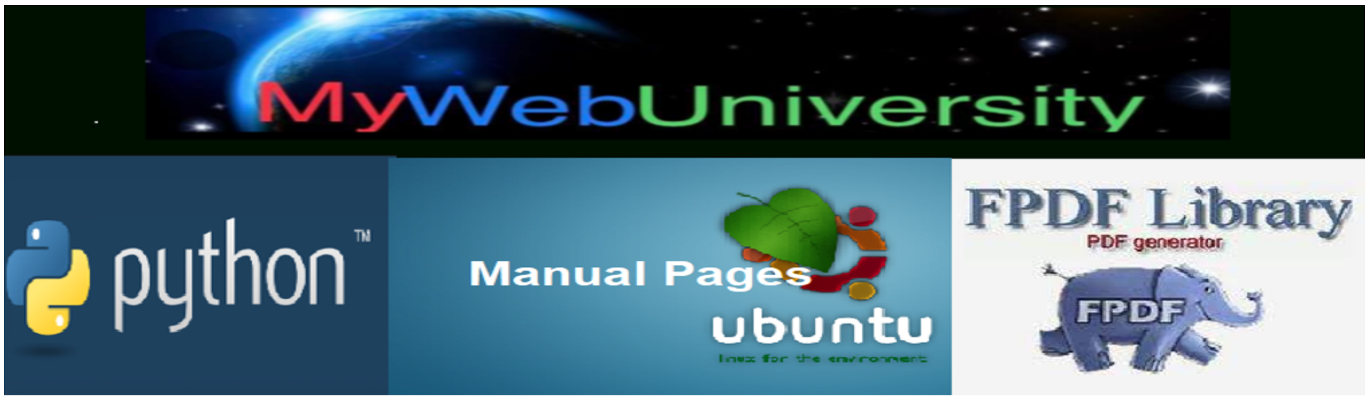

Rocky Enterprise Linux 9.2 Manual Pages on command 'nano.1'

$ man nano.1

NANO(1)   General Commands Manual   NANO(1)

NAME

nano - Nano's ANOther editor, inspired by Pico

SYNOPSIS

nano [options] [[+line[,column]] file]...

nano [options] [[+[crCR](/|?)string] file]...

DESCRIPTION

nano is a small and friendly editor. It copies the look and feel of Pico, but is free software, and implements several features that Pico lacks, such as: opening multiple files, scrolling per line, undo/redo, syntax coloring, line numbering, and soft-wrapping overlong lines.

When giving a filename on the command line, the cursor can be put on a specific line by adding the line number with a plus sign (+) before the filename, and even in a specific column by adding it with a comma. (Negative numbers count from the end of the file or line.) The cursor can be put on the first or last occurrence of a specific string by specifying that string after +/ or +? before the filename. The string can be made case sensitive and/or caused to be interpreted as a regular expression by inserting c and/or r after the + sign. These search modes can be explicitly disabled by using the uppercase variant of those letters: C and/or R. When the string contains spaces, it needs to be en?closed in quotes. To give an example: to open a file at the first occurrence of the word "Foo", you would do:

```
nano +c/Foo file
```

As a special case: if instead of a filename a dash (-) is given, nano will read data from standard input.

EDITING

Entering text and moving around in a file is straightforward: typing the letters and using the normal cursor movement keys. Commands are entered by using the Control (^) and the Alt or Meta (M-) keys. Typing ^K deletes the current line and puts it in the cutbuffer. Consecutive ^Ks will put all deleted lines together in the cutbuffer. Any cursor movement or executing any other command will cause the next ^K to overwrite the cutbuffer. A ^U will paste the current contents of the cutbuffer at the current cursor position.

When a more precise piece of text needs to be cut or copied, you can mark its start with ^6, move the cursor to its end (the marked text will be highlighted), and then use ^K to cut it, or M-6 to copy it to the cutbuffer. You can also save the marked text to a file with ^O, or spell check it with ^T^T.

On some terminals, text can be selected also by holding down Shift while using the arrow keys. Holding down the Ctrl or Alt key too will increase the stride. Any cursor movement without Shift being held will cancel such a selection.

The two lines at the bottom of the screen show some important commands; the built-in help (^G) lists all the available ones. The default key bindings can be changed via a nanorc file -- see nanorc(5).

NOTICE

Since version 4.0, nano by default:

? does not automatically hard-wrap lines that become overlong,

? includes the line below the title bar in the editing area,

? does linewise (smooth) scrolling.

If you want the old, Pico behavior back, you can use --breaklonglines, --emptyline, and --jumpyscrolling (or -bej for short).

OPTIONS

-A, --smarthome

Make the Home key smarter. When Home is pressed anywhere but at the very beginning of non-whitespace characters on a line, the cursor will jump to that beginning (ei? ther forwards or backwards). If the cursor is already at that position, it will jump to the true beginning of the line.

-B, --backup

When saving a file, back up the previous version of it, using the current filename suffixed with a tilde (~).

-C directory, --backupdir=directory

Make and keep not just one backup file, but make and keep a uniquely numbered one every time a file is saved -- when backups are enabled (-B). The uniquely numbered files are stored in the specified directory.

-D, --boldtext

For the interface, use bold instead of reverse video. This will be overridden by setting the options titlecolor, statuscolor, keycolor, functioncolor, numbercolor, and/or selectedcolor in your nanorc file. See nanorc(5).

-E, --tabstospaces

Convert each typed tab to spaces -- to the number of spaces that a tab at that po? sition would take up.

-F, --multibuffer

Read a file into a new buffer by default.

-G, --locking

Use vim-style file locking when editing files.

-H, --historylog

Save the last hundred search strings and replacement strings and executed commands, so they can be easily reused in later sessions.

-I, --ignorercfiles

Don't look at the system's nanorc nor at the user's nanorc.

-J number, --guidestripe=number

Draw a vertical stripe at the given column, to help judge the width of the text. (The color of the stripe can be changed with set stripecolor in your nanorc file.)

-K, --rawsequences

Interpret escape sequences directly, instead of asking ncurses to translate them. (If you need this option to get some keys to work properly, it means that the ter? minfo terminal description that is used does not fully match the actual behavior of your terminal. This can happen when you ssh into a BSD machine, for example.) Us? ing this option disables nano's mouse support.

-L, --nonewlines

Don't automatically add a newline when a text does not end with one.   (This can cause you to save non-POSIX text files.)

-M, --trimblanks

Snip trailing whitespace from the wrapped line when automatic hard-wrapping occurs or when text is justified.

-N, --noconvert

Disable automatic conversion of files from DOS/Mac format.

-O, --bookstyle

When justifying, treat any line that starts with whitespace as the beginning of a paragraph (unless auto-indenting is on).

-P, --positionlog

For the 200 most recent files, log the last position of the cursor, and place it at that position again upon reopening such a file.

-Q "regex", --quotestr="regex"

Set the regular expression for matching the quoting part of a line.   The default value is "^([\t]*([!#%:;>|}]|//))+".   (Note that \t stands for an actual Tab.)

This makes it possible to rejustify blocks of quoted text when composing email, and to rewrap blocks of line comments when writing source code.

-R, --restricted

Restricted mode: don't read or write to any file not specified on the command line.

This means: don't read or write history files; don't allow suspending; don't allow spell checking; don't allow a file to be appended to, prepended to, or saved under a different name if it already has one; and don't make backup files.   Restricted mode can also be activated by invoking nano with any name beginning with 'r' (e.g. "rnano").

-S, --softwrap

Display over multiple screen rows lines that exceed the screen's width.   (You can make this soft-wrapping occur at whitespace instead of rudely at the screen's edge, by using also --atblanks.) (The old short option, -$, is deprecated.)

-T number, --tabsize=number

Set the size (width) of a tab to number columns.   The value of number must be greater than 0. The default value is 8.

-U, --quickblank

Make status-bar messages disappear after 1 keystroke instead of after 20. Note that option -c (--constantshow) overrides this. When option --minibar or --zero is

in effect, --quickblank makes a message disappear after 0.8 seconds instead of af?
ter the default 1.5 seconds.

-V, --version

Show the current version number and exit.

-W, --wordbounds

Detect word boundaries differently by treating punctuation characters as part of a word.

-X "characters", --wordchars="characters"

Specify which other characters (besides the normal alphanumeric ones) should be considered as part of a word. When using this option, you probably want to omit -W (--wordbounds).

-Y name, --syntax=name

Specify the name of the syntax highlighting to use from among the ones defined in the nanorc files.

-Z, --zap

Let an unmodified Backspace or Delete erase the marked region (instead of a single character, and without affecting the cutbuffer).

-a, --atblanks

When doing soft line wrapping, wrap lines at whitespace instead of always at the edge of the screen.

-b, --breaklonglines

Automatically hard-wrap the current line when it becomes overlong. (This option is the opposite of -w (--nowrap) -- the last one given takes effect.)

-c, --constantshow

Constantly show the cursor position on the status bar. Note that this overrides option -U (--quickblank).

-d, --rebinddelete

Interpret the Delete and Backspace keys differently so that both Backspace and Delete work properly. You should only use this option when on your system either Backspace acts like Delete or Delete acts like Backspace.

-e, --emptyline

Do not use the line below the title bar, leaving it entirely blank.

-f file, --rcfile=file

Read only this file for setting nano's options, instead of reading both the system-wide and the user's nanorc files.

-g, --showcursor

Make the cursor visible in the file browser (putting it on the highlighted item) and in the help viewer. Useful for braille users and people with poor vision.

-h, --help

Show a summary of the available command-line options and exit.

-i, --autoindent

Automatically indent a newly created line to the same number of tabs and/or spaces as the previous line (or as the next line if the previous line is the beginning of a paragraph).

-j, --jumpyscrolling

Scroll the buffer contents per half-screen instead of per line.

-k, --cutfromcursor

Make the 'Cut Text' command (normally ^K) cut from the current cursor position to the end of the line, instead of cutting the entire line.

-l, --linenumbers

Display line numbers to the left of the text area. (Any line with an anchor addi? tionally gets a mark in the margin.)

-m, --mouse

Enable mouse support, if available for your system. When enabled, mouse clicks can be used to place the cursor, set the mark (with a double click), and execute short? cuts. The mouse will work in the X Window System, and on the console when gpm is running. Text can still be selected through dragging by holding down the Shift key.

-n, --noread

Treat any name given on the command line as a new file. This allows nano to write to named pipes: it will start with a blank buffer, and will write to the pipe when the user saves the "file". This way nano can be used as an editor in combination with for instance gpg without having to write sensitive data to disk first.

-o directory, --operatingdir=directory

Set the operating directory. This makes nano set up something similar to a chroot.

-p, --preserve

Preserve the XON and XOFF sequences (^Q and ^S) so they will be caught by the ter?
minal.

-q, --indicator

Display a "scrollbar" on the righthand side of the edit window. It shows the posi?
tion of the viewport in the buffer and how much of the buffer is covered by the
viewport.

-r number, --fill=number

Set the target width for justifying and automatic hard-wrapping at this number of
columns. If the value is 0 or less, wrapping will occur at the width of the screen
minus number columns, allowing the wrap point to vary along with the width of the
screen if the screen is resized. The default value is -8.

-s "program [argument ...]", --speller="program [argument ...]"

Use this command to perform spell checking and correcting, instead of using the
built-in corrector that calls hunspell(1) or spell(1).

-t, --saveonexit

Save a changed buffer without prompting (when exiting with ^X). (The old form of
the long option, --tempfile, is deprecated.)

-u, --unix

Save a file by default in Unix format. This overrides nano's default behavior of
saving a file in the format that it had. (This option has no effect when you also
use --noconvert.)

-v, --view

Just view the file and disallow editing: read-only mode. This mode allows the user
to open also other files for viewing, unless --restricted is given too.

-w, --nowrap

Do not automatically hard-wrap the current line when it becomes overlong. This is
the default. (This option is the opposite of -b (--breaklonglines) -- the last one
given takes effect.)

-x, --nohelp

Don't show the two help lines at the bottom of the screen.

-y, --afterends

Make Ctrl+Right and Ctrl+Delete stop at word ends instead of beginnings.

-z, --suspendable

Obsolete option; ignored. Suspension is enabled by default, reachable via ^T^Z.

(If you want a plain ^Z to suspend nano, add bind ^Z suspend main to your nanorc.)

-%, --stateflags

Use the top-right corner of the screen for showing some state flags: I when auto-
indenting, M when the mark is on, L when hard-wrapping (breaking long lines), R
when recording a macro, and S when soft-wrapping. When the buffer is modified, a
star (*) is shown after the filename in the center of the title bar.

-_, --minibar

Suppress the title bar and instead show information about the current buffer at the
bottom of the screen, in the space for the status bar. In this "minibar" the file?
name is shown on the left, followed by an asterisk if the buffer has been modified.
On the right are displayed the current line and column number, the code of the
character under the cursor (in Unicode format: U+xxxx), the same flags as are shown
by --stateflags, and a percentage that expresses how far the cursor is into the
file (linewise). When a file is loaded or saved, and also when switching between
buffers, the number of lines in the buffer is displayed after the filename. This
number is cleared upon the next keystroke, or replaced with an [i/n] counter when
multiple buffers are open. The line plus column numbers and the character code are
displayed only when --constantshow is used, and can be toggled on and off with M-C.
The state flags are displayed only when --stateflags is used.

-0, --zero

Hide all elements of the interface (title bar, status bar, and help lines) and use
all rows of the terminal for showing the contents of the buffer.   The status bar
appears only when there is a significant message, and disappears after 1.5 seconds
or upon the next keystroke. With M-Z the title bar plus status bar can be toggled.
With M-X the help lines.

-!, --magic

When neither the file's name nor its first line give a clue, try using libmagic to
determine the applicable syntax.

TOGGLES

Several of the above options can be switched on and off also while nano is running.   For
example, M-L toggles the hard-wrapping of long lines, M-S toggles soft-wrapping, M-N tog?
gles line numbers, M-M toggles the mouse, M-I auto-indentation, and M-X the help lines.

See at the end of the ^G help text for a complete list.

The M-X toggle is special: it works in all menus except the help viewer and the linter.

All other toggles work in the main menu only.

FILES

When --rcfile is given, nano will read just the specified file for setting its options and
syntaxes and key bindings. Without that option, nano will read two configuration files:
first the system's nanorc (if it exists), and then the user's nanorc (if it exists), ei?
ther ~/.nanorc or $XDG_CONFIG_HOME/nano/nanorc or ~/.config/nano/nanorc, whichever is en?
countered first. See nanorc(5) for more information on the possible contents of those
files.

See /usr/share/nano/ and /usr/share/nano/extra/ for available syntax-coloring definitions.

NOTES

If no alternative spell checker command is specified on the command line nor in one of the
nanorc files, nano will check the SPELL environment variable for one.

In some cases nano will try to dump the buffer into an emergency file. This will happen
mainly if nano receives a SIGHUP or SIGTERM or runs out of memory. It will write the buf?
fer into a file named nano.save if the buffer didn't have a name already, or will add a
".save" suffix to the current filename. If an emergency file with that name already ex?
ists in the current directory, it will add ".save" plus a number (e.g. ".save.1") to the
current filename in order to make it unique. In multibuffer mode, nano will write all the
open buffers to their respective emergency files.

BUGS

The recording and playback of keyboard macros works correctly only on a terminal emulator,
not on a Linux console (VT), because the latter does not by default distinguish modified
from unmodified arrow keys.

Please report any other bugs that you encounter via:

https://savannah.gnu.org/bugs/?group=nano.

When nano crashes, it will save any modified buffers to emergency .save files. If you are
able to reproduce the crash and you want to get a backtrace, define the environment vari?
able NANO_NOCATCH.

HOMEPAGE

https://nano-editor.org/

SEE ALSO

nanorc(5)

/usr/share/doc/nano/ (or equivalent on your system)